

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of
Inventor(s): SJÖBLOM



Appln. No.: 09 | 903,863
Series ↑ | ↑ Serial No.
Code

Group Art Unit: 2661

Filed: July 13, 2001

Examiner: Not Yet Assigned

Title: CONTROLLED DATA NETWORK ERROR RECOVERY

Atty. Dkt. P 281544 | 290051US/HS/HER
M# Client Ref

Date: September 7, 2001

**SUBMISSION OF PRIORITY
DOCUMENT IN ACCORDANCE
WITH THE REQUIREMENTS OF RULE 55**

Hon. Asst Commissioner of Patents
Washington, D.C. 20231

Sir:

Please accept the enclosed certified copy(ies) of the respective foreign application(s) listed below for which benefit under 35 U.S.C. 119/365 has been previously claimed in the subject application and if not is hereby claimed.

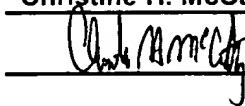
<u>Application No.</u>	<u>Country of Origin</u>	<u>Filed</u>
990102	FINLAND	January 19, 1999

Respectfully submitted,

Pillsbury Winthrop LLP
Intellectual Property Group

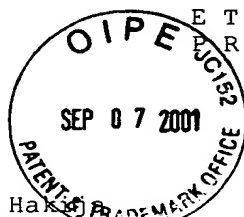
1600 Tysons Boulevard

McLean, VA 22102
Tel: (703) 905-2000
Atty/Sec: CHM/JRH

By Atty: Christine H. McCarthy Reg. No. 41844
Sig:  Fax: (703) 905-2500
Tel: (703) 905-2143

PATENTTI- JA REKISTERIHALLITUS
NATIONAL BOARD OF PATENTS AND REGISTRATION

Helsinki 4.7.2001



Hakija
Applicant

Nokia Telecommunications Oy
Espoo

Patenttihakemus nro
Patent application no

990102

Tekemispäivä
Filing date

19.01.1999

Kansainvälinen luokka
International class

H04L

Keksinnön nimitys
Title of invention

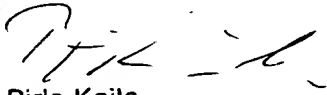
"Controlled data network error recovery"
(Ohjattu tietoverkon toipuminen virhetilanteessa)

Hakijan nimi on hakemusdiaariin 05.12.1999 tehdyn nimenmuutoksen jälkeen **Nokia Networks Oy**.

The application has according to an entry made in the register of patent applications on 05.12.1999 with the name changed into **Nokia Networks Oy**.

Täten todistetaan, että oheiset asiakirjat ovat tarkkoja jäljennöksiä patentti- ja rekisterihallitukselle alkuaan annetuista selityksestä, patenttivaatimuksista, tiivistelmästä ja piirustuksista.

This is to certify that the annexed documents are true copies of the description, claims, abstract and drawings originally filed with the Finnish Patent Office.


Pirjo Kaila
Tutkimussihteeri

Maksu 300,- mk
Fee 300,- FIM

Osoite: Arkadiankatu 6 A Puhelin: 09 6939 500 Telefax: 09 6939 5328
P.O.Box 1160 Telephone: + 358 9 6939 500 Telefax: + 358 9 6939 5328
FIN-00101 Helsinki, FINLAND

Controlled data network error recovery

Background of the invention

The present invention relates to method and equipment for transmission both in fixed networks and mobile networks, e.g. GPRS backbone
5 networks and W-CDMA backbone networks.

In the packet switched network systems a mechanism called a sliding window is used to control the flow of packets across a data link. As each packet is transmitted, the upper window edge UWE is incremented by unity. Similarly, as each packet is acknowledged, the lower window edge LWE is incremented by unity/acknowledged packet. The sending of new packets is
10 stopped, when the difference between UWE and LWE becomes equal to the size of send window. Then the sending node retries to send these sent but not acknowledged packets to the same receiving node. The sending node is a packet data transmission node, which can generate packets and transfer
15 packets other nodes have generated. If the receiving node then receives re-sent packet(s) correctly, it can determine if a received packet is a valid transmitted packet or a duplicate in this simple case where only two nodes are involved. Determining is usually done by comparing the sequence number of received packet to sequence numbers of successfully delivered packets. Normally the sequence number inserted into each packet by the sending node.
20

A problem with the current solution arises when the receiving node is "dead" for some time. In real live environments several kind of network failures may occur or data transmitting elements may go down due to a network element failure. When that happens also the resending of packets to the same
25 node fails. Then the sending node reroutes sending and sends packet (or packets) to another node via which it can route packets to the end system. However, it is possible, that the first node got that packet (or packets) and has send it (them) forward before failure. The sending node does not know when the failure happened. It does not know whether the failure happened when the
30 packet was sent first time or when the packet was received or has only the response (acknowledgement) got lost. Therefore duplicates are sent to the end system. The end system has to check for every packet it receives whether it is a duplicate for example in order not to bill a customer twice. One possible way to solve this problem is not to send duplicates, but then important information
35 may be lost.

Same problems are encountered in the systems using frames, packets or any other resendable unit. A frame comprises usually protocol related header and payload data. An empty frame is a frame with no payload data.

5 Disclosure of the invention

The object of the invention is to overcome the above stated problems. The object of the invention is achieved with a method, a system and network nodes which are characterized in that which is disclosed in the independent claims. The preferred embodiments of the invention are set forth in
10 the dependent claims.

The invention is based on indicating that this unit is possibly a duplicate to that entity which it is resent because no response is received from the entity it was sent.

The advantages of the invention are that the possible duplicates are
15 indicated, so that the system can handle these units differently than other units. For example, the end system or some other system does not need to check for every unit whether it is a duplicate. This minimizes the load in the system. The load is also minimized since the removal of duplicated information from the data received eases the unit handling load.

In one embodiment of the invention also the sending entity is indicated. The advantage of this embodiment is, that it is possible to use as an intermediate storage for different sending entities one and same receiving entity, since the units may be identified with sending unit identity and sequence
20 number. Therefore they are not mixed with other units having same sequence number in the receiving entity and the network operation. A further advantage is that network maintenance does not need to make sure that two sending entities do not use same receiving entity as the intermediate storage.
25

In one embodiment of the invention the unit indicates to be a possible duplicate unit is sent forward from the second entity only after the sender
30 has given instructions to send. Before giving instructions the sender has checked from the first entity whether it has got the packet. The advantage of this embodiment is that the duplicate unit is not sent in vain in the network and thus the network load is minimized. Another advantage of this embodiment is that there is no risk to get duplicates of units caused by the communication
35 failure between the sending node and receiving entity. Yet another advantage

of this embodiment is that the load due to crosschecking of duplicates in a border area (e.g. a month has changed in the billing system) can be avoided.

In one embodiment of the invention the possibility that a unit is a duplicate is checked in the end system only when the received unit has been indicated to be a possible duplicate. The advantage of this embodiment is that the duplicate checking is not done in vain and the end systems can be made more simple. Besides the units arrive to the end so that it is possible that the order can be re-established.

Brief description of the figures

10 The invention will be described in further detail in the following by means of preferred embodiments with reference to the accompanying drawings, in which

Figure 1 is a flow chart illustrating the functionality of a sending entity in the first preferred embodiment;

15 Figure 2 is a flow chart illustrating the functionality of a receiving entity in the first preferred embodiment;

Figure 3 is a flow chart illustrating the functionality of a sending entity in the second preferred embodiment;

20 Figure 4 is a flow chart illustrating the functionality of an end system according to the invention; and

Figure 5 illustrates one example of a system according to the invention.

Detailed description of the invention

25 The present invention is suitable for use both in fixed communications systems and in mobile communications systems. The invention is particularly suitable for use in implementing the General Packet Radio Service (GPRS) in the pan-European digital mobile communications system GSM (Global System for Mobile Communications) or corresponding mobile communications systems, such as DCS1800 and PCS (Personal Communication System). The invention is suitable also for third generation mobile systems, such as Universal Mobile Communication System (UMTS) and Future Public Land Mobile Telecommunication System (FPLMTS) later renamed as IMT-2000 (International Mobile Telecommunication 2000), which at present are being developed. The present invention may be used e.g. in handovers when
35 the frames are resent also to the target network entity.

The present invention can be implemented to the existing network nodes. They all have processors and memory with which the inventive functionality described below may be implemented. In some embodiments some extra memory may be needed. The functions described below may be located
5 in one network element or some of them may be in one element and the others in other elements regardless how they are located in the examples used to illustrate the invention. These transmitting nodes are also called intermediate nodes. Since this sending illustrated e.g. in figures 1-4 may be an internal exchange of information, nodes are also called entities. The term 'node' as used
10 herein should be understood to generally refer to any network element or functionality which handles the units.

In the following description term packet is used in the sake of clarity. The term packet should be understood to mean also any other resendable unit e.g. a frame. Frames are used e.g. in the radio link protocol. In the following
15 the invention is described using two intermediate entities for the sake of clarity yet without limiting the invention to that kind of solutions. It is even possible to use the invention where only one receiving node exists and the sending node can not reroute resendable units when the sending node has enough memory. The preferable embodiments have, however, at least two alternative directions
20 to send these resendable units from the sending node.

In the following description it is assumed for the sake of clarity, that the packet generating node has difficulty with the first node via which it tries to send one packet to the exit node, but it has no difficulty with the second node. The end system, e.g. a billing system, is described here to be one entity, al-
25 though the end system may comprise several different entities. The structure of the end system is not important to the invention. It is also assumed that only one packet is sent for the sake of clarity. The person skilled in the art knows how to deal with a plurality of packets being possible duplicates, that is how to handle e.g. a window, which size is bigger than one.

30 Figure 1 illustrates the functionality of a sending node (entity) in the first preferred embodiment of the present invention. The sending node can be a packet generating node (entity) or an intermediate node sending the packet further ahead. In step 101 node sends a packet P1 to entity 1. The entity 1 may be the primary peer. The packet P1 has a sequence number with which it
35 can be identified within a window size. The window size must be smaller than the max sequence number in order to identify the packet properly (unless the

window size is one). The node discovers in step 102 that no response is received. In step 103 it tries to resend the packet P1 to the entity 1, but fails. In another embodiment the IPD is sent in step,103. Resend means here the same as retry send. It is done configured amount of times after time outs of
 5 determined lengths. Then it stores into its memory the sequence number SN of P1 and the entity 1 in step 104. That's how it knows where it sent packet P1 without response. Then, in step 105 the node picks from it priority list an entity which priority is smaller, but closest to the priority of entity 1. With picking is meant that the sending node selects according to predetermined rules the next
 10 entity (or its address) to which it will try to send next. That entity is here entity 2. After that it stores in step 106 into its memory the entity 2 so that it knows that it has first tried to send packet P1 to entity 1 and after that to entity 2. Then it adds an IPD to the packet P1 in step 107. The IPD is an indication indicating possible duplicate of the packet P1. After that the node sends in step
 15 108 the packet P1 to the entity 2 and receives a positive response from it in step 109. Positive response means here acknowledgement of receiving packet P1.

If the positive response is not received from entity 2, the node can repeat the steps 105 - 109 until a positive response is received. With positive
 20 response it is meant that the packet P1 was received successfully. As long as the entity 1 is "dead", the packets may be send via the entity 2.

Then in step 110 the node notices, that entity 1 is again "alive", that is packets can be send to it again. That, how the node notices this entity 1 being alive is described later with alternative examples described in more de-
 25 tail with references to figure 5. The node checks in step 111 from its memory, whether there are packets sent to entity 1, which may have a duplicate. That is it checks e.g. whether in its buffer for unconfirmed packets are packets. If there are no, it continues normally either sending new packets via entity 1 or entity 2 depending of the configuration. In this example the serving node finds
 30 out, that packet P1 is a possible duplicate and sends a test packet to entity 1 in step 112. This test packet is an empty packet with the sequence number of P1.

After that the node receives a response from entity 1 and checks in step 113 is the response ok. If the response is ok, the entity 1 never got the
 35 right packet P1 or did not succeed to send it. The response is ok, if it is e.g. a request accepted. Therefore there are no duplicates and the node sends in

step 114 to the entity 2 a message indicating that entity 2 can release packet P1. In other words node allows the entity 2 to send the packet P1 ahead. If in step 113 the response indicates that entity 1 has already received that packet, the node sends in step 115 to the entity 2 a message indicating that the entity
 5 2 can delete packet P1 from its memory. Also a cancel message can be used for the same purpose. In other words the node 2 is not allowed to send the packet P1 ahead. The packet P1 is identified in message send either in step 114 or 115 by the sequence number. It is also possible to use some identification of entity 1 in messages send in steps 108, 114 and 115. The message
 10 which indicate that the response is not ok may e.g. be a request already fulfilled. The sending of new packets to entity 1 is preferably done after instructions on all unconfirmed packets are sent.

In some other embodiments instead of storing done in steps 104 and 105 buffering can be used, but then there is a risk of losing information
 15 due to a failure.

In some other embodiments the whole packet P1 may be saved in step 104 and send as a test packet in step 112. It is yet possible to save the packet only in the entity 2 and when sending test packet the sending node first asks the entity 2 to send a copy of it. Depending from the application in these
 20 embodiments, the message send in step 114 may only allow to release those packets which were not send as a test packet and the test packet is deleted because its sequence number was not in the release message.

If the resend does not fail in step 103, further packets are sent normally to the entity 1.

25 Figure 2 illustrates the functionality of a packet receiving entity in the first preferred embodiment. The receiving entity is assumed to be an intermediate entity. The receiving entity RE receives in step 201 a packet P1 from a sending entity SE. The RE checks in step 202 whether there is an IPD (indication of possible duplicate) in the packet P1. If there is, the RE stores in
 30 step 203 the packet P1 in order to wait for instructions from the SE. It may also store the packet P1 with the information, that it received it from the SE and/or some identification of the first entity if indicated by the SE. In step 204 the RE waits for instructions. During this waiting it may transmit other packets normally. In step 205 the RE receives instructions concerning packet P1 from the
 35 SE. (It identifies the packet P1 e.g. from the sequence number.) In step 206 it checks indicated the instruction a delete or cancel. If it was a cancelling or

deleting instruction, the RE deletes from its memory the packet P1 in step 207. If the instruction did not indicate a delete/cancel but a release, the RE sends the packet P1 ahead normally in step 208. In the first preferred embodiment the packet P1 has still the IPD with added information that it is released by an
 5 instruction from the sending entity, so that e.g. the end system may still check whether it has got it earlier, but the other intermediate nodes do not store it to wait instructions since it has this added information with the IPD. In some other embodiments the RE may take the IPD away from the packet. Then no other checking of duplicates is done in vain. If the packet is released in em-
 10 bodiments using max storing time before delivery because of the length of the storing time of this packet reached this storing time, it is very advantageous to have the IPD in the packet.

If there were no IPD in step 202, the RE sends the packet P1 ahead normally in step 208.

15 Figure 3 illustrates the functionality of a sending node (entity) in the second preferred embodiment of the present invention. In step 301 node sends a packet P1 to entity 1. The node discovers in step 302 that no response is received. Then, in step 303 the node picks from its address list an address of the next entity, entity 2. Then it adds IPD (an indication indicating
 20 possible duplicate of the packet P1) to the packet P1 in step 304 and sends in step 305 the packet P1 to entity 2. In some other embodiments the step 303 may be similar to the step 105 of figure 1 and/or between steps 302 and 303 the step 103 of figure 1 may be done. It is assumed that after step 305 a positive acknowledgement (response) is received. If not, then the steps 303-305
 25 are repeated until a positive response is received.

The packet P1 is sent ahead in the intermediate entities (nodes) in the second preferred embodiment until it reaches the end system. Figure 4 illustrates the functionality of an end system in the second preferred embodiment of the invention. In embodiments where only one receiving node exists,
 30 the functionality described in figure 4 may be implemented into it. When an embodiment related to the first embodiment of the invention is used, where the receiving entity does not remove the IPD when sending a packet forward, the end system does not need to function like illustrated here with figure 4.

Referring to figure 4 the end system ES receives a packet P1 in
 35 step 401 and checks in step 402 whether there is an IPD (indication of possible duplicate) in the packet P1. If there is, it goes through all the packets it has

received in step 403 in order to find out whether it has already received this packet P1. If the ES finds out in step 404 that the packet P1 is a duplicate, it deletes in step 405 the packet P1 with IPD it received in step 401. If the ES finds out in step 404 that it has not received the packet P1, in other words the
 5 packet P1 is not a duplicate, it saves in step 406 the packet P1 or at least enough information in order to do duplicate check if needed. Then it sends the packet P1 or its information to further processing according to normal procedures which depends on the application. If in step 402 no IPD is found, the ES goes right to the step 406.

10 The first preferred embodiment suits very well to those applications where the order of packets in the end system is not important, like billing systems or email. Its advantage is that the network is not loaded unnecessary by sending duplicates through whole network. The second preferred embodiment may be used also with systems where the order of the packets is of some im-
 15 portance, like still pictures or photos.

 The steps have not been set out in absolute time sequence in figures 1, 2, 3 and 4. Some of the above described steps may take place simultaneously or in different order or some of the steps can be skipped over, e.g. steps 110-115. It is also possible to add new steps not shown in figures, for
 20 example in figure 1 between steps 109 and 110 new packets can normally been sent to entity 2 without marking them as duplicates. Another example is that it is possible to check before step 203 in figure 2 whether there is with IPD added information that packet is released by an instruction from the sending entity, and if there is go to step 208, otherwise go to step 203. It is also possi-
 25 ble to combine steps from figures when making a new embodiment. For example, it is possible to further process the packet in step 406 by taking the IPD away and send the packet ahead. Essential is, that the possibility of the packet being a duplicate is indicated. The indication can be done e.g. by adding it to the packet header or to the payload or by sending a message indicating that
 30 the following packet(s) is (are) possible duplicates or in the file name when file protocols are used. It may also be in another frame. It is also possible to indicate the duplicate with the message the unit is send, e.g. send packet means that packet is not a duplicate whereas send possibly duplicated packet indicates a possible duplicate. The indication may even go via another link. It is
 35 not important how this indication is done, essential is that it is done. The messages may include more information than what is stated above. The names of

the messages may differ from those set out above or the indications or the instructions according to the invention may be sent in other messages as stated above. For example delete may be cancel or the IPD may be called mark of potential/possible duplicate MPD.

5 In the above storing means that the information is stored so that it is not lost e.g. during a restart. In other words it is stored to a non volatile memory. The information may be stored in the sending unit, and/or any other living entity with which the sending unit has a connection. That entity may be the receiving entity or totally different entity. In the above a sequence number is
10 used to identify the packet. Also other identification may be used. In a preferably embodiment the sending unit indicates also the first receiving entity when it indicates a possible duplicate. With this the receiving entity knows whose possible duplicates it has. This is very advantageous since same intermediate entity can store possible duplicates first sent to different nodes and yet identify
15 them properly. It is possible to indicate the sending node and use this information for identifying purposes.

Figure 5 illustrates one example of a system according to the invention. For the sake of clarity, the figure 5 has only one Packet Generating Node PGN 1, although it is possible to have a plurality of PGNs. The PGN 1 has in
20 this illustrative example three links Link 1 to Packet Receiving Node PRN 1, Link 2 to Packet Receiving Node PRN 2 and Link 3 to Packet Receiving Node PRN 3. The PGN 1 can send packets to the End System ES via all Packet Receiving Nodes. The packet receiving nodes are intermediate entities. The packets are sent ahead until they reach the End System ES. Although not il-
25 lustrated in figure 5, there may be any number of PRNs between e.g. PRN 1 and ES. If the system illustrated in figure 5 is a GPRS billing system, then the PGN 1 may be a serving GPRS support node SGSN or a GPRS gateway support node GGSN, and Packet Receiving Nodes may be different nodes which have charging gateway functionality CGF. So they may be called as charging
30 gateway nodes. The End System ES may be the billing system. The GPRS billing system with one charging gateway is described in more detail in Finnish patent number 102232. This patent is incorporated herein by reference.

Below some alternative examples are described in more detail with references to figure 5. The abbreviations used are:

35 BS = Billing System

CDMA = Code Division Multiple Access

CDR = Call Detail Record

CGF = Charging Gateway Function

IMSI = International Mobile Subscriber Identity

GPRS = General Packet Radio Service

5 NE = Network Element

O&M = Operations and Maintenance

PDP = Packet Data Protocol

W-CDMA = Wideband CDMA

PGN = Packet Generating Node

10 PRN = Packet Receiving Node

ES = End System

The application areas in the following examples are environments where it is not absolutely crucial that the packet sent arrive in the exactly original order from PGNs to ES, but where it is useful that no packet contents is
 15 lost even in abnormal network link failure situations or NE failure situations. For example charging data collection in packet data based telecommunications systems is a very likely application area. One example of that kind of environment is GPRS charging. In GPRS the SGSNs and GGSNs are PGNs, the CGFs are the PRNs and the BS is the ES. Each packet transmitted be-
 20 tween a PGN and PRN may contain one or more CDRs as payload inside the packet frame.

An example architectural figure 5 of a chain of network elements, where PGN 1 sends packets towards ES via either PRN 1, PRN 2 or PRN 3. The PRN 1 is here assumed to be the primary choice (priority 1 PRN peer
 25 name configured to its PRN address list as the first place to attempt packet sending). Packet information contents flow is assumed to be following : Packet Generating Node(s) -> Packet Receiving Nodes -> End System

Packet receiving node has mass memory for packets and also the end system has mass memory for packets.

30 Preparatory assumptions are following:

- The topmost protocol in the communication software stack that transfers packets between the PGN and the PRNs is assumed to be a Request-Response type message based protocol.

- Packet send window size per each link from packet generating
 35 node is smaller than the max sequence number (that is allowed to run over back to 0 and increase again per each packet).

- In most telecommunication systems like in these examples, the mass storage devices can be assumed to maintain their information even when the network element would go down due to a software failure or lack of processing capacity in relation to the traffic load.

5 Possible problems which are solved here in these examples are following:

- Duplicated information (e.g. packets containing CDRs) might be generated when traffic is redirected and packet generating node does not surely know if the redirected packets were already successfully transmitted to
10 the packet receiving node 1 or not.

- A network operator is governed by the laws and administration of the country within it operates. The operators are audited by officials. An operator might be subjected to penalties or even lose its network operator licence if it would generate too big bill(s) to its subscriber(s), because of duplicated CDR information.
15

- Also, it could lose its credibility among its customers if some user data packets or CDRs related to them are either duplicated or lost.

- On the other hand, operators do not want to unnecessarily lose money, so they do not want to cancel unnecessarily packets containing CDRs
20 unless they are 100% sure the packets are duplicates.

Example A

Having sequence numbers for packets (typically in the frame of the packets) in each link from packet generating node (PGN) to packet receiving node (PRN). The sequence numbers can be incremented by one onwards and
25 roll over again to 0 after e.g. 65535, but important is that the max sequence number is bigger than the max receive window size of a PRN.

Redirection of packets (marked with a Potential Duplicate flag) to a parallel node PRN 2 (or PRN 3 if PRN 2 is not available for some reason, etc.) in case PRN 1 fails. PGN1 send also to PRN2 some identification information
30 of PGN1, so that the PRN2 can identify these packets. This is necessary since PRN2 can have packets marked with a Potential Duplicate flag also from other PGNs with same sequence numbers.

PRNs keep potentially marked duplicate packets in memory buffer(s) or mass storage so that the information of their origin (PGN1 or
35 PGN2 or other PGN) can also be related to the packets. This is necessary since the sequence numbers of packets are unique at a moment only in each

PGN-PRN communication link, but not necessarily unique at a time in the whole network.

Keeping track in packet generating node(s) per each packet receiving node (=per each link) of sequence numbers of packets whose successful transmission was not sure. If the PGN1 fails with sending a packet to PRN1 and also fails sending a possible duplicate of the packet to PRN2, then the PGN1 tries to send the possible duplicate of the packet to another PRN, e.g. PRN3. However, there is no need to keep track of sequence numbers of those possible duplicates which were tried to send to PRN2 per this link, since these packets are already kept track per link to PRN1. It is, however, possible to add to the information relating to link PRN1, that these packets are sent also to PRN2 and PRN3.

Sensing by PGN(s) when peer node(s), PRN(s), come(s) again alive. Either PGNs send keep alive messages to PRNs after appropriate intervals (getting echoing response back from PRNs if the PRNs are alive and working ok), or PRNs (and possibly other nodes too) can inform their peer nodes (configured communication partners) always when after being non-working they come alive. Also when a node is stopping working, e.g. when the operator wants to stop it to make a software update, the node can inform its peer nodes that it is going down and should not any more receive packets before it announces to become alive again.

Sensing what information the previously collapsed node had been able to process successfully.

Informing the secondary receiving node which packets it can send forward towards the end system.

It might also occur that at some time the PGN node goes down and its volatile memory contents is destroyed, including the buffer for sequence numbers of packets whose reception by e.g. PRN 1 was not possibly successful and who have been sent (marked as potential duplicates) to a PRN 2 to wait for a later decision by the PGN. In this case the potentially duplicates might stay very long at PRN 2. Therefore it is recommendable that PRNs have either an enough long time out to cancel themselves such packets or the operator is provided tools for getting information about this kind of situation and a tool to delete such long waiting potential duplicates by an O&M operation from a PRN 2. Another alternative is to allow finally the sending of those packets (e.g. marked as potential duplicates) towards the ES.

Example B

Similar as example A) but the sensing method for finding out whether the PRN 1 (who had gone down and then become alive again) has successfully handled already a packet (sent by PGN 1) related to a sequence number is different. Here the PGN 1 would send a normal whole packet again to PRN 1, the only difference being that now the packet is marked a potential duplicate. This requires that the intermediate storage for the whole packets could reside either in PGN1, or the whole (potentially duplicated) packet should first be fetched from PRN 2 (where it has been waiting in an intermediate storage for a final decision whether it can be sent to the ES). The PGN 1 could ask PRN 2 to give a specific potentially duplicated packet back by referring to the sequence number of the packet, and after getting back the potentially duplicated packet from PRN 2, it could be resent to PRN 1. After that PRN 1 would give response to PGN 1 (whether it has already serviced that kind of sending (i.e. processed successfully that packet or whether it got the packet "for first time")). Then there are two possible submechanisms in case it was "new" to PRN 1: B1) PRN 1 processes such packet towards the ES and informs PGN 1 and PGN 1 cancels the potentially duplicated packet stored as backup in PRN 2 (or even PGN 1), or B2) PRN 1 keeps potentially duplicated packets in its intermediate memory for potential duplicates as long as PGN 1 makes the selecting commands (to cancel the packet in PRN1 and release towards ES the potentially duplicated packet stored in PRN 2, or vice versa).

Advantages of examples

Performance increase in the receiving end system, since either it has to do a check for duplicated packets (containing e.g. CDRs) for only the very small minority of packets that have been produced in an abnormal case of a network node or link failure and marked as potential duplicates or no packets produced are duplicated in the PGN-PRN interface even in abnormal node or link failure events.

Another advantage is that reliability increases, regarding to the information contents received.

Yet another advantage is reduction of possible manual error recovery procedures.

The examples presented here do not require that the receive window sizes of the PRN in a network are the same, so the O&M events that con-

figure the window size are easier to accomplish in practise, as all the receive window sizes of the PRNs need not to be updated simultaneously.

It is possible to use as an intermediate storage for different PGNs one and same PRN, since the packets are identified at least with PGN and sequence number. Therefore they are not mixed with other packets in the PRN and the O&M does not need to make sure that two PGNs do not use same PRN as the intermediate storage.

Important Features described in examples are:

1) The two here presented sensing methods in examples A) and B) that the PGN can use to find out whether the PRN has successfully received and processed those packets that it received before the Link 1 or PRN become malfunctioning.

2) The buffering of sequence numbers of packets sent from PGN to PRN 1, related to not successfully confirmed packets sent by the PGN 1. This buffer is maintained in each PGN for each PRN that the PGN is allowed to be connected to.

3) The buffering of sequence numbers of possibly duplicated packets sent to a PRN 2 to wait for a later decision (Cancel or Release) by the PGN 1.

4) The similar redundancy method, but with the difference that the buffers for potential duplicates for each PGN 1 -> PRN x resides in PGN 1 itself.

5) The idea of marking (e.g. to packet frames) the potential duplicates, allowing them to be handled differently than the other packets (e.g. to wait in some node for final confirmation whether they are allowed to be sent to the ES or whether those packets should be cancelled i.e. deleted). This is the only essential feature for this invention.

6) The idea of identify the packets in the intermediate PRN using identification of PGN and a sequence number.

The accompanying drawings and the description pertaining to them are only intended to illustrate the present invention. Different variations and modifications to the invention will be apparent to those skilled in the art, without departing from the scope and spirit of the invention defined in the appended claims.

Claims

1. A method in a telecommunications system where a sending entity may send units to a first receiving entity; the method comprising the steps of:

sending an unit to the first receiving entity;

5 receiving no response from said first receiving entity;

characterized by:

indicating a possible duplication of said unit when resending it.

2. A method according to claim 1, characterized by the method further comprising the step of indicating also the sending entity when
10 indicating said possible duplication.

3. A method according to claim 1 or 2, characterized in that the possible duplicate is indicated unit when resending said unit to the second receiving entity.

4. A method according to claim 3, characterized by the
15 method further comprising the steps of:

noticing that the first receiving entity is operating;

checking whether the first receiving entity received said unit; and

sending a release message to the second receiving entity when
said unit was not received in the first receiving entity; or

20 sending a cancel message to the second receiving entity when said unit was received in the first receiving entity.

5. A method according to claim 3, characterized by

the method further comprising the steps of:

noticing that the first receiving entity is operating;

25 checking whether the first receiving entity received said unit by resending said unit; and

sending a release message or cancel message to the second receiving entity when said unit was not received in the first receiving entity; or

30 sending a cancel message to the second receiving entity when said unit was received in the first receiving entity.

6. A method according to claim 4 or 5, characterized by the method further comprising the steps of:

receiving said unit in the second receiving entity;

storing said unit in a response of said indication; and

35 sending said unit in a response to said release message from the second receiving entity towards its destiny; or

deleting said unit in a response to said cancel message.

7. A method according to any of the preceding claims, characterized by the method further comprising the steps of:

receiving said unit in its end system;

5 checking only in a response of said indication whether the unit is a duplicate.

8. A method according to any of the preceding claims, characterized by the method further comprising the step of indicating the possible duplication by adding said indication to the unit before resending it.

10 9. A transmission system comprising a sending entity (PGN1) and at least one receiving entity (PRN1, PRN2, PRN3),

characterized in that

said sending entity (PGN1) is arranged when not receiving a response from a first receiving entity, to which it sent a unit, to indicate a possible duplication of said unit when resending it.

15 10. A system according to claim 9, characterized in that said sending entity (PGN1) is further arranged to indicate also the sending entity (PGN1) when indicating said possible duplication.

11. A system according to claim 9 or 10, characterized in that the system is arranged to indicate said possible duplication when resending said unit to a second receiving entity (PRN1, PRN2, PRN3).

12. A system according to claim 9, 10 or 11, characterized in that the receiving entity (PRN1, PRN2, PRN3) is arranged to check from a received unit whether it includes said indication and in response to said indication to wait for instructions on how to handle said unit.

13. A system according to claim 9, 10 or 11, further comprising an end system (ES) characterized in that the end system (ES) is arranged to check from a received unit whether it includes said indication and only in response to said indication to check whether said unit is a duplicate.

14. A network node (PGN1) which is adapted to be a sending entity in a network, which node (PGN1) is arranged to send an unit to a first receiving entity (PRN1, PRN2, PRN3), characterized in that the node is arranged when not receiving a response from the first entity (PRN1, PRN2, PRN3) to which it sent a unit to indicate that said unit is a possible duplication when resending said unit.

15. A network node as claimed in claim 14 characterized in that the node (PGN1) is further arranged to indicate the sending entity (PGN1) when indicating said possible duplication.

5 16. A network node as claimed in claim 14 or 15, characterized in that the node is further arranged to indicate said possible duplication when resending said unit to another entity (PRN1, PRN2, PRN3).

17. A network node as claimed in claim 16 characterized in that the node (PGN1) is further arranged to have a priority list of entities to which it may send units and the node is arranged to send the unit to the entity
10 having the next lowest priority.

18. A network node (ES) which is adapted to be a part of end system in a network, characterized in that the node is arranged to check only in a response of a received unit having an indication indicating a possible duplication of said unit, whether the unit is a duplicate.

15 19. A network node (PRN1, PRN2, PRN3) which is adapted to be an intermediate node in a network, characterized in that the node (PRN1, PRN2, PRN3) is arranged to check when receiving an unit whether it is indicated to be a possible duplication of said unit and in a response to said indication to wait for instructions on how to handle said unit.

(57) Abstract

A method, a system and network nodes using indication of the possible duplicates (IPD) of units, so that these units can be handled differently than other units. The unit is indicated to be a possible duplicate to that entity to which it is resent because no response is received from the entity it was sent.

(Figure 3)

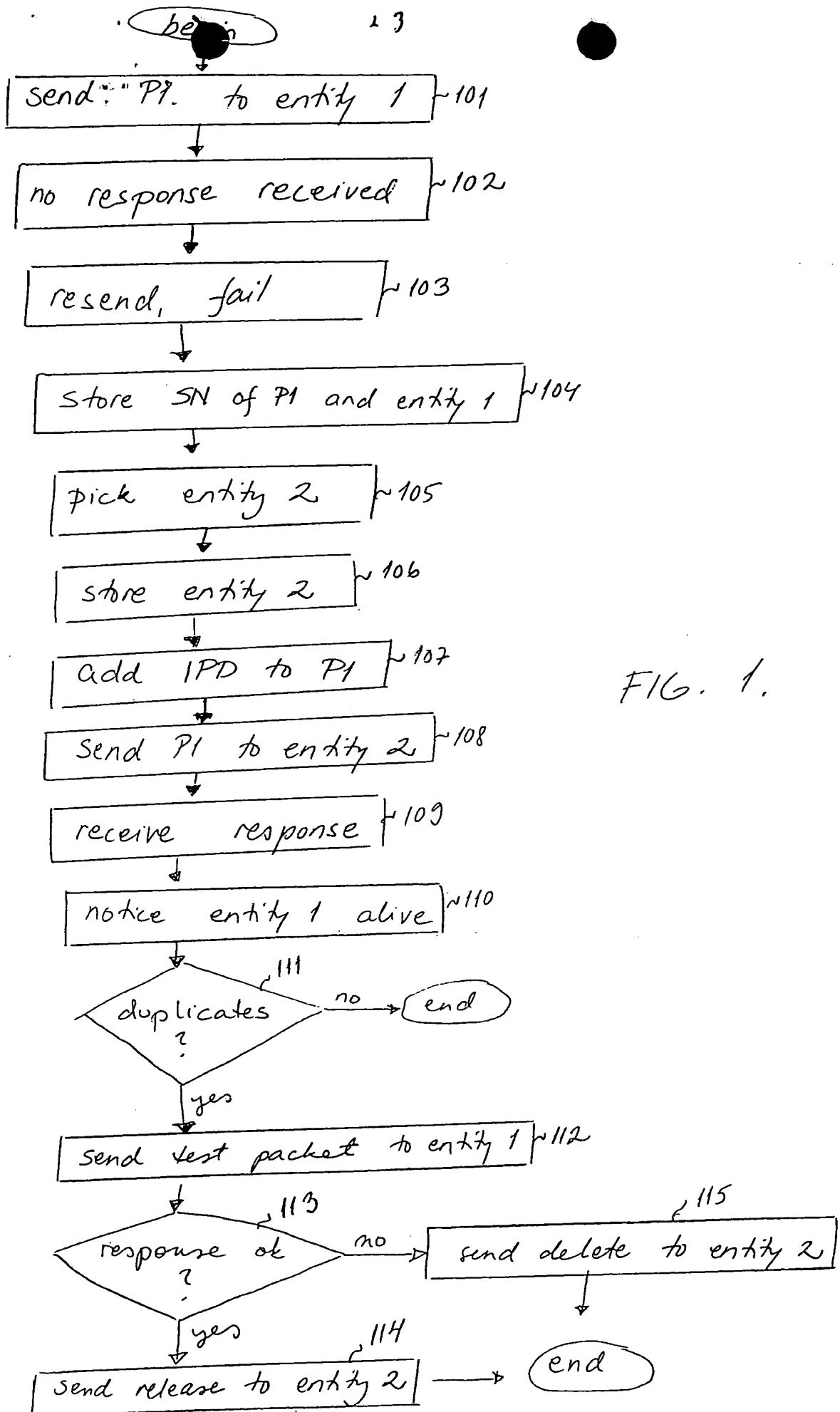


FIG. 1.

